Accelerating the Training of Convolutional Neural Networks for Image Segmentation with Deep Active Learning

Weitao Chen, Rick Salay, Sean Sedwards, Vahdat Abdelzad and Krzysztof Czarnecki

Abstract— **Semantic segmentation is an important perception function for automated driving (AD), but training a deep neural network for the task using supervised learning requires expensive manual labelling. Active learning (AL) addresses this challenge by automatically querying and selecting a subset of the dataset to label with the aim to iteratively improve the model performance while minimizing labelling costs. This paper presents a systematic study of deep AL for semantic segmentation and offers three contributions. First, we compare six different state-of-the-art querying methods, including uncertainty-estimate, Bayesian, and out-of-distribution methods. Our comparison uses the state-of-the-art image segmentation architecture DeepLab on the Cityscapes dataset. Our results demonstrate subtle differences between the querying methods, which we analyze and explain. We show that the differences are nevertheless robust by reproducing them on architecture-independent randomly generated data. Second, we propose a novel way to aggregate the output of a query, by counting the number of pixels having acquisition values above a certain threshold. Our method outperforms the standard averaging approach. Finally, we demonstrate that our findings remain consistent for whole images and image crops.**

I. Introduction

Semantic image segmentation (SIS) is an important perception task in robotics and automated driving (AD). Recently, deep supervised learning has shown promising results in this task. However, deep supervised learning requires a large amount of labelled data. labelling images for SIS by human annotators is both expensive and time-consuming relative to data collection [1].

Active Learning (AL) [2] addresses this problem by selecting only a subset of collected data to label. Given a set of unlabelled data, AL aims to find a small subset that gives the most accurate model [3]. The process can be iterative, where each query is based on the model learned so far, hence the name *active* learning. Thus, the cost of labelling is minimized while the performance is maximized.

Currently, active learning for SIS with deep neural networks has not been extensively studied. Most works in this area focus on image classification [4]–[8], while the few that do focus on SIS with deep learning have limited querying methods and outdated network architectures [9]–[11].

This paper contributes a comparison of multiple querying methods for their effectiveness in AL. We also propose a novel way to query images by counting the number of pixels with acquisition values above a certain threshold. Our results are then compared between querying whole images and image crops. The predictor used is DeepLabv3+ [12], a state-of-the-art deep neural network for SIS. The dataset is Cityscapes [1], a collection of finely annotated street-view images captured from a vehicle in cities.

Following a brief background and related work on SIS and AL, we describe the methods used in this research, including each query method, network architecture, and training process. Finally, we present and discuss the experimental results.

Authors are with the University of Waterloo, Waterloo, Ontario, Canada.
This work is funded by Japanese Science and Technology agency (JST) ERATO project JPMJER1603: HASUO Metamathematics for Systems Design, by Natural Sciences and Engineering Research Council of Canada (NSERC) CREATE in Product-Line Engineering for Cyber-physical Systems (PLoCS) grant and by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant: Model-Based Synthesis and Safety Assurance of Intelligent Controllers for Autonomous Vehicles.

II. Background and Related Work

A. Semantic Image Segmentation

SIS [13] is the process of classifying each pixel in a given image. The classes are predefined and each pixel must be labelled as one of them. An exception is the "ignore" class. Pixels labelled as "ignore" in the ground truth will not affect evaluation. A common metric for evaluating SIS is the mean Intersection-Over-Union (mIOU) [14].

B. Active Learning

AL is described in detail in the survey by Settles [15]. In pool-based AL [16], which is our focus, a pool of data is collected but not labelled. AL queries the unlabelled pool to select samples to be labelled by an oracle, who is usually a human annotator. The newly labelled data are then added to the labelled training set. Using the training set, a machine learning model is trained through supervised learning. The model is then evaluated on the validation set. If the performance is satisfactory or if the budget for labelling is spent, the process is stopped. If the stopping criteria are not met, the model is used to query new samples from the unlabelled pool and the whole cycle starts again (see Fig. 1).

There are many different ways to query. The simplest one is random querying. To perform better than the random baseline, a querying method typically examines each datum and uses an acquisition function to estimate some notion of information gain. Since one wants to maximize the information gain, data with the largest acquisition values are often chosen. In some cases, however, it might be beneficial to query part of data with lower acquisition values to diversify the samples. Fu, Zhu, and Li [17] have a good example of the effects of sample diversity in their paper.

C. Active Learning for Semantic Segmentation

AL for SIS has similar ideas as AL for classification. One difference is that since each image has multiple pixel

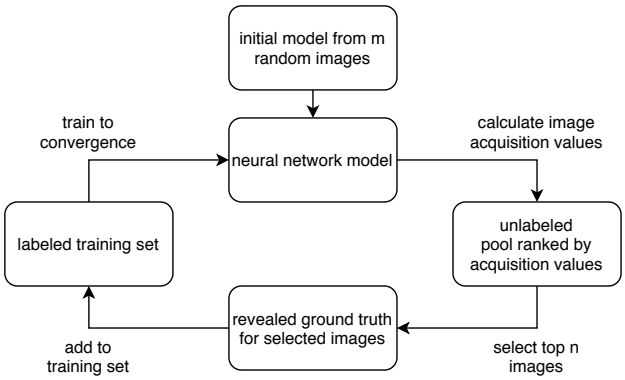

Fig. 1. Active learning cycle

predictions, the labelling cost for each image could be different, and crops of images could be queried in region-based AL [9], [18]. In this paper, however, we assume the labelling cost for each image is the same and images must be cropped before the first AL cycle begins.

AL for SIS with deep learning is explored in [9]–[11]. Gorriz et al. [10] use U-net [19] for the network and Monte-Carlo dropout for querying. Yang et al. [11] use Fully Convolutional Network (FCN) [20] for network and bootstrapping [21] as acquisition function with cosine similarity for sample diversity. Mackowiak et al. [9] also use FCN for the network and explored entropy querying and vote entropy querying.

III. Research Questions and Methodology

Our research objective is to examine and compare different AL designs for SIS. More specifically, it is to answer the question of what is the best way to label a pool of data to train a deep model to achieve the highest performance with the lowest labelling cost. To answer this question, we perform experiments that explore the AL design space, which consists of three main dimensions: queried unit type, pixel acquisition function, and aggregation method. Each dimension has multiple choices, and thus the best AL design is a combination of choices from each design dimension.

Fig. 2 illustrates our experiment design tree. For each AL experiment, a querying unit can be either a batch of whole images or image crops. In this paper, the word "image" refers to both whole image and image crop, unless stated otherwise. The querying order can be either random (our baseline) or ranked by an image acquisition function. Each image acquisition value is aggregated from its pixel acquisition values either by averaging or by counting values (our proposal) over a threshold. We explore 6 different pixel acquisition functions in addition to random selection. Overall, the diagram represents 26 different querying approaches.

A. Acquisition Functions

The acquisition functions compared are random, entropy [22], max-softmax [23], margin [24], ODIN [25], BALD [26], and vote entropy [27]. They are either standard, common, or state of the art in relevant contexts. Random is a standard baseline query; entropy, max-softmax and margin are common queries in active learning [15]; ODIN is a state-of-the-art calibration technique; BALD and vote entropy are state-of-the art uncertainty measures.

1) Random: Random querying is used as the baseline for all comparisons. Each image in the unlabelled pool has an equal probability to be queried. The acquisition function is a pseudo-random number generator of uniform distribution.

2) Entropy: The closer the distribution is to uniform, the more uncertain the model is, and the higher entropy it has [22]:

$$S = -\sum_c P_c \log P_c$$

where P_c is the softmax output of the neural network for class c.

3) Max-softmax: It considers the probability of the most likely class as a measure of confidence [23]. We pick the samples that are the least confident:

$$S = 1 - \max_c(P_c)$$

4) Margin: It measures the ambiguity between the top two classes by taking the difference between the highest two class probabilities P_1 and P_2 [24]:

$$S = 1 - (P_1 - P_2)$$

5) ODIN: ODIN (Out-of-Distribution detector for Neural networks) [25] is an enhancement of max-softmax, where the softmax function is scaled by temperature T. T is a hyperparameter that can be calibrated for a neural network model to give a more accurate confidence prediction [28]. The calibrated softmax function is

$$P_i(\boldsymbol{x}; T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{c=1}^{N} \exp\left(f_c(\boldsymbol{x})/T\right)}$$

where $f_c(x)$ is the output of the neural network before softmax for class c. N is the number of classes. P_i is the calibrated probability for class i. The input image x is also preprocessed by adding a small perturbation scaled by ε, such that

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \varepsilon \,\text{sign}(-\nabla_{\boldsymbol{x}} \log P_{\hat{y}}(\boldsymbol{x}; T))$$

and where

$$P_{\hat{y}}(\boldsymbol{x}; T) = \max_c P_c(\boldsymbol{x}; T).$$

The per-pixel acquisition function for ODIN querying follows (III-A.3).

6) BALD: BALD (Bayesian Active Learning by Disagreement) [26] is a Bayesian AL technique for image classification [4]. In this approach, a Bayesian network uses dropout [29], which is normally a stochastic regularization technique used during training. In BALD, dropout is performed during training as well as during inference [30]. The result is a committee of different models from the same network. The per-pixel acquisition function for BALD querying is given by

$$\mathbb{I}[y, \boldsymbol{\omega} | \mathbf{x}, \mathcal{D}_{\text{train}}] = \mathbb{H}[y | \mathbf{x}, \mathcal{D}_{\text{train}}] - \mathbb{E}_{P(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})}\left[\mathbb{H}[y | \mathbf{x}, \boldsymbol{\omega}]\right].$$

$\mathbb{H}[y|\mathbf{x}, \mathcal{D}_{\text{train}}]$ is the entropy of marginal posterior y given input $\mathbf{x}$ and pool data $\mathcal{D}_{\text{train}}$ (y is approximated as an average over the committee member outputs [4]).

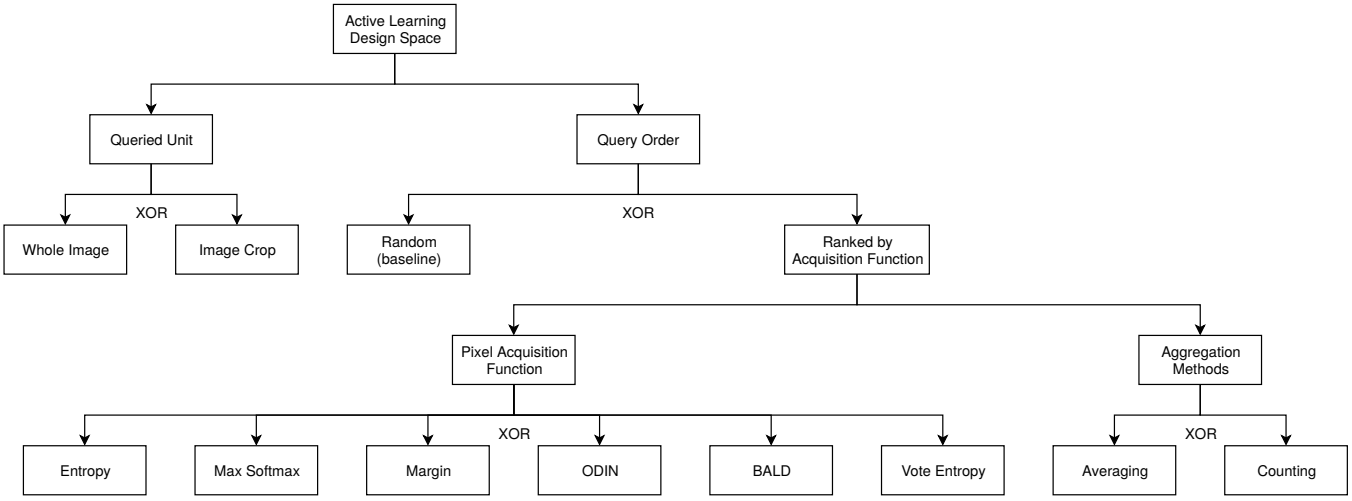

Fig. 2. Experiment design tree

7) Vote Entropy: Vote Entropy [27] applies a Bayesian network and MC dropout, similar to BALD. The per-pixel acquisition function for vote entropy querying is

$$V := -\sum_{c} \frac{\sum_{e} D(P_e, c)}{N_E} \cdot \log \frac{\sum_{e} D(P_e, c)}{N_E}$$

$$\text{where} \quad D(a, c) = \begin{cases} 1 & \text{if } \mathit{argmax}(a) = c \\ 0 & \text{otherwise.} \end{cases}$$

Variable e represents a member of committee (sampled through dropout), and N_E is the number of models in the committee. P_e is the softmax output of member e.

B. Aggregation Methods

In order to evaluate an image, all its pixel acquisition values need to be aggregated into a single image acquisition value. Choosing a good aggregation method is challenging because it is difficult to know the importance of a pixel before labelling.

A simple and reasonable assumption is that each pixel has importance in proportion to its acquisition value, such that the image acquisition value is simply the average of all pixel acquisition values. The image acquisition value of aggregation by averaging is thus given by

$$S_{\text{image}} = \sum_{\text{pixel} \in \text{image}} S_{\text{pixel}} / N_{\text{image}}$$

where S_{pixel} is the pixel acquisition value, and N_{image} is the number of pixels in the given image.

An alternative view is that not all pixels are useful. Hence, we propose a novel aggregation method that counts the number of pixels that have a pixel acquisition value greater than some threshold a. This gives an estimate of the area of informative regions. The image acquisition value of aggregation by counting is given by

$$S_{\text{image}} = \sum_{\text{pixel} \in \text{image}} D(S_{\text{pixel}}, a) / N_{\text{image}}$$

$$\text{where} \quad D(s, a) = \begin{cases} 1 & \text{if } s > a \\ 0 & \text{otherwise.} \end{cases}$$

C. Querying Unit

Mackowiak et al. [9] show that querying image crops outperforms querying whole images with the same total number of pixels. Since this has been done, we do not replicate their experiment with region-based AL. Instead, we experiment with different querying and aggregation methods on a pool of random crops. For every whole image in the dataset, we randomly cropped a single 512×512 crop, 8 times smaller than the whole image, and discarded the rest of the image. The collection of random crops are now treated as the new data pool. During every AL cycle, each crop is treated as a whole image, albeit with different resolution, and everything else is kept the same for comparison. The evaluations are still performed on the original validation set with whole images.

The difference between our approach and region-based AL is that it uses a sliding window to scan all the whole images. Crops within the sliding window are selected by a metric, and every selected crop is labelled. Thus, cropping becomes part of the image querying process. In our approach, the image querying process comes after cropping.

IV. Experiment Setup

A. Network Architecture

The network architecture we use in this work is DeepLabv3+, which extends DeepLabv3 with encoder-decoder structure [12]. It employs techniques from previous versions, including atrous convolution and atrous spatial pyramid pooling (ASPP) [31]–[33]. Currently, DeepLabv3+ has state-of-the-art results on SIS datasets, including Cityscapes [12].

The backbone architecture we use is resnet_v1_50_beta, which modifies ResNet-101 [34] by replacing the first 7×7 convolution with three 3×3 convolutions [12]. The weights are pretrained on ImageNet [35].

During training, we use a batch size of 8, a base learning rate of 0.007, weight decay of 0.0001, atrous rates of 6, 12, and 18, an output stride of 16, and a decoder output stride of 4. For data preprocessing, every image is randomly cropped to 512×512, randomly flipped, and scaled by a factor between 0.5 and 2. For this setup, a model fully-trained on the entire Cityscapes training set can achieve around 75% mIOU.

For BALD, the dropout rate is 0.9 for training, and 0.5 for inference. The difference in dropout rates between training and inference are compensated by weight scaling. The number of models in the committee is 10 for both BALD and vote entropy.

B. Training Procedure

Since the Cityscapes dataset is already fully labelled, the ground truth is simply hidden until declared labelled. This way, there is no need for a human annotator and the oracle is part of the program. The dataset is divided into 2975 images in the training set and 500 images in the validation set.

Since training for image segmentation is computationally expensive, adding labelled images one by one is not feasible. Therefore, a batch of size $n = 50$ is queried in every AL cycle. This is not to be confused with the training batch size. For image crops, since each crop is eight times smaller than a whole image, we make the querying batch size eight times larger $(n = 400)$, so that each query acquires the same number of pixels. To ensure all querying methods have a fair comparison, every experiment starts with the network trained to convergence with the same $m = 50$ images, selected as follows. The network was fully trained twice with each of five sets of 50 randomly selected images, giving a total of ten training runs and corresponding models. The performance of the ten models ranged from 50% mIOU to 52% mIOU. We selected a model having performance around 51% mIOU to be the initial model for all experiments, i.e. a model in the middle of the range.

Fig. 1 shows our experimental AL cycle. The current trained model is used to query the unlabelled pool, in order to select the next batch of images with the highest acquisition values. In the first cycle, the common initial model is used to query the unlabelled pool. The selected images have their ground truth revealed and are added to the labelled training set. Using the combined training set, the model is trained again from scratch to avoid catastrophic forgetting. The cycle repeats until the stopping criteria are met.

V. Results

Our experiments are divided into three parts, each corresponding to a dimension of the experiment design shown in Fig. 2. The first part examines all of the pixel acquisition functions with aggregation by averaging on whole images. We compare the three uncertainty measures: entropy, max-softmax, and margin querying, then the two Bayesian methods: BALD and vote entropy. ODIN is studied separately to demonstrate the effect of temperature T and perturbation ε. For the second part, the most promising methods from the first part are selected to apply aggregation by counting. In the final part, aggregation by counting is compared with averaging on image crops. All experiments are compared with the random baseline as reference.

To evaluate a given querying method, we look at the learning curve, which plots the model's mIOU on the validation set as a function of the cumulative amount of labelled data in the training set, as the AL cycles proceed. Since the goal is to save labelling costs, the best querying method has a learning curve that rises the quickest. Eventually, all querying methods converge to the performance of a model trained on the entire data pool. All of our learning curves are averaged over four independent runs. We find that the variance between runs is small. To compare methods, we use area under the learning curve (ALC), where each learning curve is normalized w.r.t. the maximum mIOU possible with its corresponding data pool. In what follows, we use the term ALC to refer to this normalized metric. Since the fastest growing learning curve has the highest ALC, a higher ALC indicates a better querying method when the learning curves do not intersect.

We use our results to answer three questions. First, whether using uncertainty and OOD measures for querying improves the performance of AL over the random baseline, and how these measures perform relative to one another. Second, how our proposed counting-based image acquisition value aggregation compares to the existing average-based aggregation. Third, whether the answers to the previous questions hold when the queried units are whole images as well as when the units are image crops.

Our results are summarized in Fig. 3, where the average learning curves for each method are normalized by scaling them w.r.t. the mIOU of the first cycle and the mIOU of the entire corresponding data pool. Normalization is needed because there are eight times fewer pixels in the image crop data pool: each crop is eight times smaller than the whole image, with the regions outside crops not in the data pool. The maximum achievable unnormalized mIOU for our particular selection of random crops is 65%, compared with 75% for whole images. All normalized curves eventually reach 100%, but we focus on the range 50 to 400 images, which includes the entire data pool for crops and is sufficient to illustrate the differences in performance. Since the learning curves are close and occasionally overlap, for clarity we plot the range of the active querying methods as a shaded band. The random baselines are plotted with solid lines.

A. Aggregation by Averaging on Whole Images

Our first experiment compares all of the considered pixel acquisition functions, using aggregation by averaging on whole images (orange curves in Fig. 3). We can see immediately that all pixel acquisition functions perform better

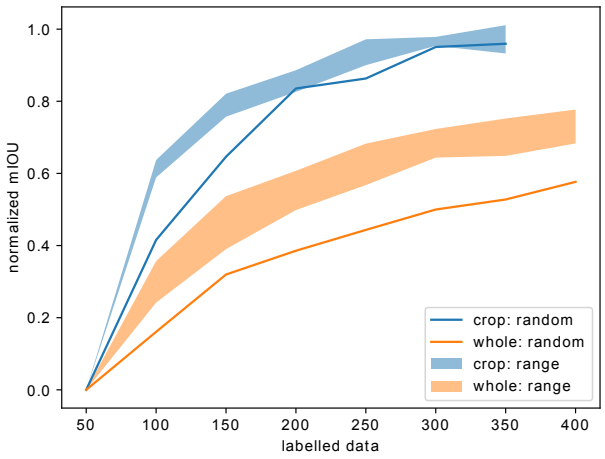

Fig. 3. Summary of all normalized learning curves. The x-axis represents the number of whole images or their equivalent amount of pixels. For example, 200 on the x-axis represents 200 whole images and 1600 image crops because they have the equivalent amount of pixels. The y-axis represents the normalized mIOU where 0 maps to the real mIOU of the initial model and 1 maps to the mIOU where the whole dataset is queried.

TABLE I
NORMALIZED ALC FOR WHOLE IMAGES

Counting	Querying Method	ALC
	Random (baseline)	146
	ODIN: $T = 1000$, $\varepsilon = 0.0014$	186
	ODIN: $T = 10$	193
	ODIN: $T = 100$	202
	ODIN: $T = 1000$	203
	ODIN: $T = 0.01$	206
	Margin	207
	Vote Entropy	208
	ODIN: $T = 0.1$	209
	Max Softmax	211
✓	Margin: $a = 0.8$	214
✓	Max Softmax: $a = 0.5$	215
	Entropy	215
	BALD	218
✓	**Entropy: $a = \log(2)$**	**222**

than the random baseline at every AL cycle. There is also a significant gap between the band of querying methods and the random baseline. This shows that using any of the querying methods achieves a higher mIOU than using random querying, for any amount of labelled data. For example, at 250 whole images, the top querying outperforms random querying by 6% mIOU. From an alternative perspective, using any querying methods can save labelling cost at any mIOU level. For example, at 63% mIOU, the top querying method uses 50% less labelled data than random querying.

The differences between the active querying methods are more subtle, especially between the best methods. These tend to be concentrated near the top of the shaded band. To distinguish the methods, Table I lists the normalized ALC of all querying methods considered. The querying methods marked with ✓, and that specify threshold a, use aggregation by counting, while the rest use aggregation by averaging. We see that the ALCs of the querying methods with aggregation by average are close to each other, ranging from 186.4 to 221.7. On the other hand, random querying is much lower at 145.7.

The ALC values show that all active querying methods are better than random querying, and that the differences between them are small. These small differences may seem surprising, given that the acquisition functions have very different properties and purposes. One plausible explanation is that our experimental architecture is close to the optimal performance of AL, given the dataset and given the querying units are whole images. It may not be possible to achieve a significantly higher ALC.

We now look at different groups of querying methods in more detail.

The three uncertainty measures are entropy, max-softmax, and margin, which we note have similar ALC performance. Although these three methods give different emphasis to the range of softmax distributions they encounter, they agree exactly at the extremes [15]. A uniform distribution has the maximum pixel acquisition value in all three measures, while a single peak distribution has the minimum in all three. We therefore hypothesize that it is these extremes that dominate the performance of the uncertainty metrics in the present context.

The two Bayesian methods, vote entropy and BALD, show small but significant differences. We observed that vote entropy querying performed no better than entropy querying in our experiments, which is in agreement with the result from Mackowiak et al. [9]. However, BALD performed better than vote entropy and entropy in every AL cycle, making it one of the best querying methods by a small margin. This is possibly because BALD gives relatively low acquisition values for the pixels around the contour of objects and the boundary where one class transitions into the next. These contour pixels have high aleatoric uncertainty [36], which may not be beneficial.

ODIN querying is a more complex case. Liang, Li, and Srikant [25] show that as T increases, OOD detection performance increases monotonically, with the improvement diminishing as T becomes large. We find the intuition of this result is incorrect in the context of active learning. Noting that max-softmax is equivalent to ODIN with $T = 1$ and $\varepsilon = 0$, our experiments show that the ascending order of performance for T is $10, 100, 0.1, 1$, which is not monotonic. Liang, Li, and Srikant also show that a large T and a moderate epsilon is the best for OOD detection, with $T = 1000$ and $\varepsilon = 0.0014$ working well for them. We tried these parameters in our active learning experiments, but they performed significantly worse than max-softmax. We conclude that in the context of active learning, T must be optimized as a hyperparameter.

B. Validation on Synthetic Data

The results for ODIN are surprising, so we performed additional experiments on synthetic data to eliminate the possibility that they are specific to our chosen dataset or network. Figs. 4 and 5 compare the entropy distributions of image acquisition functions for real and synthetic pixel

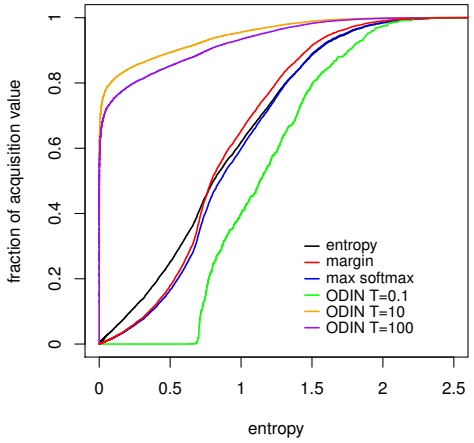

Fig. 4. Entropy distributions of real acquisition values

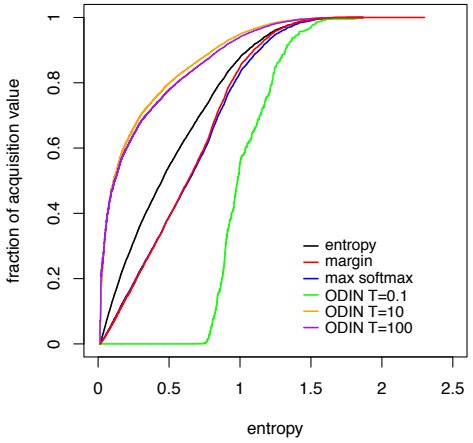

Fig. 5. Entropy distributions of synthetic acquisition values

data. The real data are the softmax distributions produced by our network for a typical image chosen in an active learning cycle. For the synthetic data, we randomly generated a pool of softmax distributions, skewed towards low entropy and approximately ranging over the same entropy values as the real data. In each case, we ordered the distributions by entropy and calculated cumulative sums of the various pixel acquisition functions. The final sum for each query method corresponds to the image acquisition value. Since the image acquisition functions do not all range over the same values, the cumulative sums were normalized. The y-axes in Figs. 4 and 5 thus report the fraction of the image acquisition value.

The figures reveal that the real and synthetic distributions are not identical, but nevertheless confirm some of the characteristics we observe in our experiments. We see in Fig. 4 that the distributions of entropy, max-softmax and margin are close, mirroring their performance in our experiments. The same curves in Fig. 5 are also close, especially in comparison to the curves for the ODIN queries. Importantly, given max-softmax is equivalent to ODIN with $T = 1$, both figures demonstrate the non-monotonic behaviour with respect to T. In both cases, the curve for ODIN $T = 10$ does not lie between the curves for $T = 1$ and $T = 100$, as might be expected, but is further away from the entropy curve than ODIN $T = 100$. These results suggest that our conclusions are robust and not dataset nor network dependent.

C. Aggregation by Counting on Whole Images

For the second part of our investigation, we selected the three uncertainty acquisition functions for aggregation by counting. For entropy querying, we experimented with a threshold $a = -2 * 0.5 * \log(0.5) = \log(2)$. The entropy at this particular threshold corresponds to a distribution with two classes having an equal probability of 0.5, and all other classes being 0. This is an important threshold because a lower entropy is more likely to have a single distinctive peak in the class probability distribution. We find that entropy querying with aggregation by counting has higher ALC than entropy querying with aggregation by average.

Next, we compare the performance of max-softmax querying. We experimented with aggregation by counting with threshold $a = 0.5$, which corresponds to the distribution having two 0.5 class probabilities and $\log(2)$ entropy. Note, however, the two query thresholds are different for other distributions. The results show that max-softmax querying by counting with $a = 0.5$ has better performance than aggregation by average.

Finally, we compare the performance of margin querying. Margin querying with threshold $a = 0.8$ excludes distributions with a difference between the top two probabilities greater than 0.2. The results show that margin querying by counting with $a = 0.8$ performs better than aggregation by averaging.

Overall, these results show that aggregation by counting performs better than the standard aggregation by averaging for entropy, max-softmax, and margin querying.

TABLE II
NORMALIZED ALC FOR CROPS

Counting	Querying Method	ALC
	Random (baseline)	234
	Margin	254
	Max Softmax	254
	Entropy	257
✓	Entropy: $a = \log(2)$	257
✓	Margin: $a = 0.8$	257
✓	**Max Softmax: $a = 0.5$**	**259**
	ODIN: $T = 0.1$	**259**

D. Image Crops

For the final part of our investigation, we compare both aggregation methods on image crops. First, note that the normalized learning curves for image crops in Fig. 3 outperform those for whole images. Further, we see in Table II that, as with the results for whole image querying, all aggregation by counting queries outperform their averaging counterparts, although by a smaller margin. We also include an experiment using ODIN with $T = 0.1$, which was one of the best performing T values. Once again, it outperforms max-softmax ($T = 1$). Overall, these experiments confirm that aggregation by counting is superior to aggregation by

averaging, regardless of whether the querying unit is a batch of whole images or image crops. The results also suggest that using image crops with aggregation by counting is the most effective query method.

VI. Conclusion

We compared six different querying methods in the context of active learning for image segmentation and found small but discernible differences among them. We demonstrated these results on the industry-standard dataset Cityscapes, as well as on randomly generated data, using the state-of-the-art image segmentation architecture DeepLabv3+. We also proposed a novel method, counting with threshold, to aggregate the pixel acquisition value. We showed our method performs better than the standard aggregation by averaging. These findings were repeated with image crop as the querying units, and the results still hold.

Due to the high computational costs, we leave the evaluation of other image segmentation datasets and architectures for future work. We believe other image aggregation methods with pixels weighted unequally would also be worth exploring. Furthermore, it would be interesting to see the combination of aggregation by counting with other dimensions, such as region-based active learning. We believe that such a combination could potentially outperform previous results.

References

[1] M. Cordts, M. Omran, et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. June 2016, pp. 3213–3223.

[2] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. "Active Learning with Statistical Models". In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 129–145.

[3] S. Tong and D. Koller. "Active Learning for Parameter Estimation in Bayesian Networks". In: *13th International Conference on Neural Information Processing Systems (NIPS)*. MIT Press, 2000, pp. 626–632.

[4] Y. Gal, R. Islam, and Z. Ghahramani. "Deep Bayesian Active Learning with Image Data". In: *34th International Conference on Machine Learning (ICML)*. Vol. 70. PMLR. Aug. 2017, pp. 1183–1192.

[5] A. Krishnamurthy, A. Agarwal, et al. "Active Learning for Cost-Sensitive Classification". In: *34th International Conference on Machine Learning (ICML)*. Vol. 70. PMLR. Aug. 2017, pp. 1915–1924.

[6] O. Sener and S. Savarese. "Active Learning for Convolutional Neural Networks: A Core-Set Approach". In: *6th International Conference on Learning Representations (ICLR)*. 2018.

[7] S. Vijayanarasimhan and K. Grauman. "What's It Going to Cost You?: Predicting Effort vs. Informativeness for Multi-Label Image Annotations". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2009, pp. 2262–2269.

[8] K. Wang, D. Zhang, et al. "Cost-Effective Active Learning for Deep Image Classification". In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (2016), pp. 2591–2600.

[9] R. Mackowiak, P. Lenz, et al. *CEREALS – Cost-Effective REgion-based Active Learning for Semantic Segmentation*. 2018. arXiv: 1810.09726.

[10] M. Gorriz, A. Carlier, et al. *Cost-Effective Active Learning for Melanoma Segmentation*. 2017. arXiv: 1711.09168.

[11] L. Yang, Y. Zhang, et al. "Suggestive Annotation: A Deep Active Learning Framework for Biomedical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2017*. Vol. 10435. LNCS. Springer, 2017, pp. 399–407.

[12] L.-C. Chen, Y. Zhu, et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *The European Conference on Computer Vision (ECCV)*. Sept. 2018, pp. 801–818.

[13] K.-S. Fu and J. K. Mui. "A Survey on Image Segmentation". In: *Pattern Recognition* 13.1 (1981), pp. 3–16.

[14] V. Badrinarayanan, A. Kendall, and R. Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495.

[15] B. Settles. *Active Learning Literature Survey*. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences, 2009.

[16] D. D. Lewis. "A Sequential Algorithm for Training Text Classifiers: Corrigendum and Additional Data". In: *SIGIR Forum* 29.2 (Sept. 1995), pp. 13–19.

[17] Y. Fu, X. Zhu, and B. Li. "A survey on instance selection for active learning". In: *Knowledge and Information Systems* 35.2 (2013), pp. 249–283.

[18] T. Kasarla, G. Nagendar, et al. "Region-Based Active Learning for Efficient Labeling in Semantic Segmentation". In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, pp. 1109–1117.

[19] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Vol. 9351. LNCS. Springer, 2015, pp. 234–241.

[20] J. Long, E. Shelhamer, and T. Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. June 2015, pp. 3431–3440.

[21] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Vol. 57. Monographs on Statistics and Applied Probability. Chapman and Hall/CRC, 1994, 456 pages.

[22] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423.

[23] D. Hendrycks and K. Gimpel. *A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks*. 2016. arXiv: 1610.02136.

[24] T. Scheffer, C. Decomain, and S. Wrobel. "Active Hidden Markov Models for Information Extraction". In: *Advances in Intelligent Data Analysis*. Vol. 2189. LNCS. Springer, 2001, pp. 309–318.

[25] S. Liang, Y. Li, and R. Srikant. *Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks*. 2017. arXiv: 1706.02690.

[26] N. Houlsby, F. Huszár, et al. *Bayesian Active Learning for Classification and Preference Learning*. 2011. arXiv: 1112.5745.

[27] I. Dagan and S. P. Engelson. "Committee-Based Sampling For Training Probabilistic Classifiers". In: *12th International Conference on Machine Learning (ICML)*. Morgan Kaufmann, 1995, pp. 150–157.

[28] C. Guo, G. Pleiss, et al. "On Calibration of Modern Neural Networks". In: *34th International Conference on Machine Learning (ICML)*. Vol. 70. PMLR. Aug. 2017, pp. 1321–1330.

[29] N. Srivastava, G. Hinton, et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[30] Y. Gal and Z. Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *33rd International Conference on Machine Learning (ICML)*. Vol. 48. PMLR. June 2016, pp. 1050–1059.

[31] L.-C. Chen, G. Papandreou, et al. *Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs*. 2014. arXiv: 1412.7062.

[32] L.-C. Chen, G. Papandreou, et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2017), pp. 834–848.

[33] L.-C. Chen, G. Papandreou, et al. *Rethinking Atrous Convolution for Semantic Image Segmentation*. 2017. arXiv: 1706.05587.

[34] K. He, X. Zhang, et al. "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2016, pp. 770–778.

[35] O. Russakovsky, J. Deng, et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[36] S. Depeweg, J. M. Hernández-Lobato, et al. "Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-sensitive Learning". In: *35th International Conference on Machine Learning (ICML)*. Vol. 80. PMLR. July 2018, pp. 1184–1193.